



# Linguagem SQL - View

Fundamentos Base de Dados - 2025/26

Carlos Costa

# View - Conceito

- Relação Virtual
  - Relação virtual derivada de relação(ões) base.
  - São vistas sobre dados detidos por tabelas reais.
    - Não existe um (segundo) armazenamento físico dos dados.
  - Permite manipular os dados da view.
- Utilização
  - Apresentação de dados
    - Adaptação do esquema de base de dados a diferentes aplicações.
  - Questões de segurança de dados
    - Integridade e Privacidade
  - Ferramenta de estruturação de queries mais complexas

# View - Cuidados de Utilização

- Existem opiniões diversas sobre o uso de views que vão desde recomendações de...  
total abstinência do seu uso...  
... até seu uso generalizado (excessivo).

## *Microsoft SQL Server 2008 Bible (Best Practice):*

*“Views are an important part of the abstraction puzzle; I recommend being intentional in their use. Some developers are enamored with views and use them as the primary abstraction layer for their databases. They create layers of nested views, or stored procedures that refer to views. This practice serves no valid purpose, creates confusion, and requires needless overhead. The best database abstraction layer is a single layer of stored procedures that directly refer to tables, or sometimes user-defined functions.”*

*Nota: Vamos ver mais à frente (no semestre) o que são stored procedures e user-defined functions.*

# SQL View - Criação (Definição)

```
CREATE VIEW <view_name> AS <SQL_query>
```

```
/* Exemplo1: Uma vista com o nome dos funcionários, projectos em  
que trabalham e número de horas. */
```

```
CREATE VIEW EMPLOYEE_PROJECTS AS  
    SELECT Fname, Lname, Pname, Hours  
    FROM    EMPLOYEE JOIN WORKS_ON ON Ssn=Essn  
           JOIN PROJECT ON Pno=Pnumber;
```

```
/* Exemplo2: Vista com nome do departamento, número de  
funcionários e total de salários. */
```

```
CREATE VIEW DEPT_INFO(Dept_name, No_of_emps, Total_sal) AS  
    SELECT    Dname, Count(*), Sum(Salary)  
    FROM      EMPLOYEE JOIN DEPARTMENT ON Dno=Dnumber  
    GROUP BY Dname;
```

# Utilização de Views

- Uma view pode ser utilizada como fonte de dados (similar a uma tabela normal) num conjunto de operações SQL já identificadas:
  - SELECT, INSERT, UPDATE, DELETE
- Existem duas aproximações, dependentes do SGBD:
  - query modification
    - transformação da query definida
  - view materialization
    - criação de uma tabela temporária com resultados da execução da view, sobre a qual serão executadas as operações SQL pretendidas
- Nested Views
  - views como fonte de dados de outras views

# Query Modification - Exemplo

- Imaginemos uma operação SELECT sobre a view EMPLOYEE\_PROJECTS criada anteriormente...

```
CREATE VIEW EMPLOYEE_PROJECTS AS
SELECT Fname, Lname, Pname, Hours
FROM   EMPLOYEE, PROJECT, WORKS_ON
WHERE  Ssn=Essn AND Pno=Pnumber;
```

**SELECT sobre EMPLOYEE\_PROJECTS**

```
SELECT Fname, Lname
FROM   EMPLOYEE_PROJECTS
WHERE  Pname='GalaxyS';
```

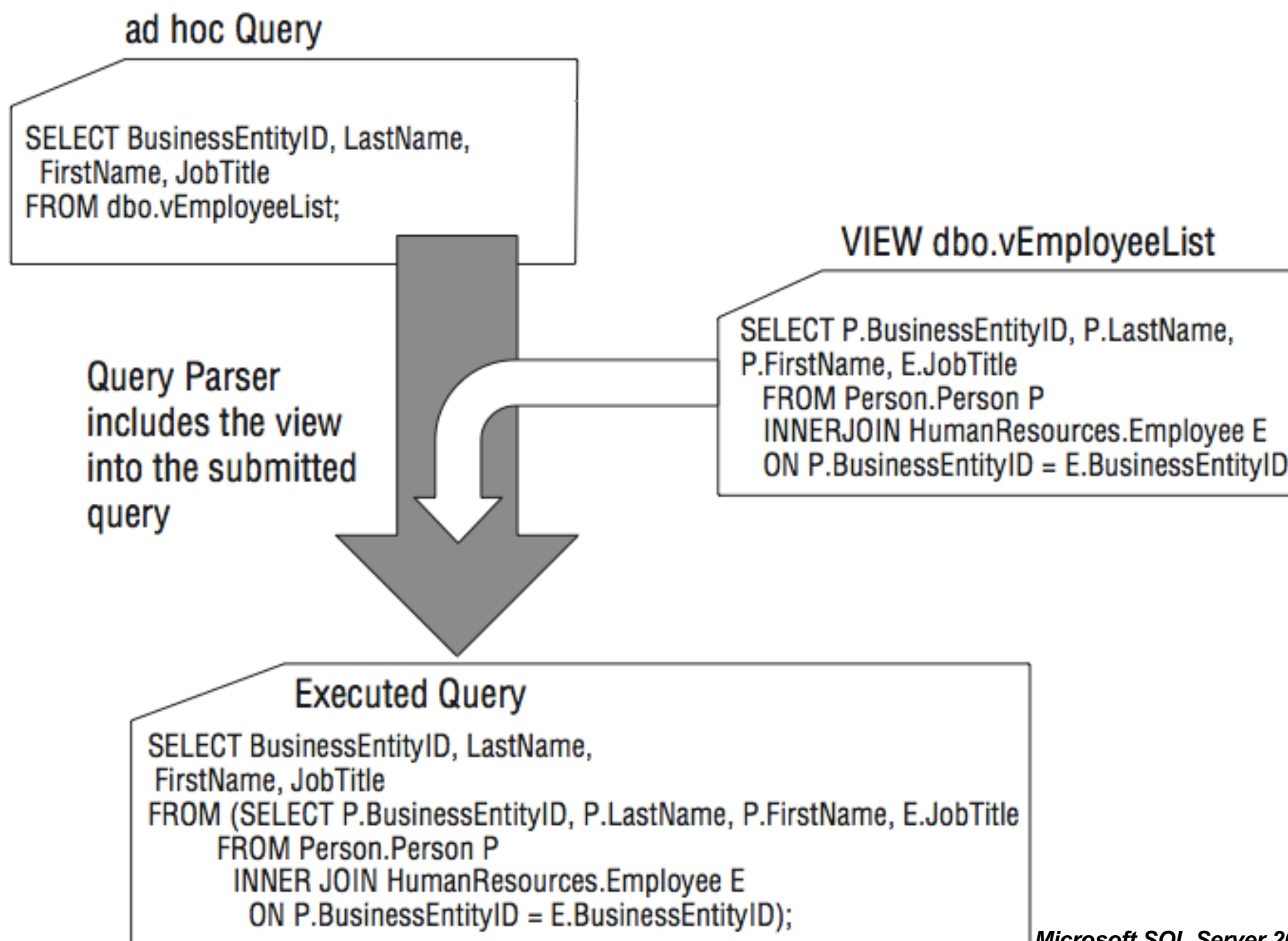


SGBD

**Query modification...**

```
SELECT Fname, Lname
FROM   EMPLOYEE, PROJECT, WORKS_ON
WHERE  Ssn=Essn AND Pno=Pnumber AND Pname='GalaxyS';
```

# Em SQL Server



# View - Alterar e Eliminar

```
CREATE VIEW <view_name> AS <SQL_query>
```

```
/* Exemplo1: alterar o vista anterior para agrupar pelo Pname,  
Hours, Fname e Lname */
```

```
ALTER VIEW EMPLOYEE_PROJECTS AS
```

```
    SELECT      Fname, Lname, Pname, Hours  
    FROM        EMPLOYEE JOIN WORKS_ON ON Ssn=Essn  
               JOIN PROJECT ON Pno=Pnumber  
    GROUP BY    Pname, Hours, Fname, Lname;
```

Alterar

```
DROP VIEW <view_name>
```

```
/* Exemplo1: Eliminar a vista EMPLOYEE_PROJECTS */
```

```
DROP VIEW EMPLOYEE_PROJECTS;
```

Eliminar

```
/* Exemplo2: Eliminar a vista DEPT_INFO */
```

```
DROP VIEW DEPT_INFO;
```

# View - Update de Dados

- Como se imagina, a actualização de dados via Views pode ser muito complexa em função da complexidade da própria view.
- Há restrições na sua utilização, dependentes do SGBD.
- Em geral, podemos dizer que uma view:
  - É updatable se incluir uma só tabela base na sua definição e os seguintes atributos: primary key e todos os NOT NULL sem default value.
  - Não é updatable se utilizar várias tabelas base (uso do join) ou utilizarem agrupamentos de atributos e funções de agregação.
- Muitos autores não recomendam este tipo de utilização.
  - muito menos em cenários de actualização de dados por parte de aplicações cliente (forms, web pages, etc).

# View - Exemplo de atualização de dados

## Definir a View...

```
CREATE VIEW EMPLOYEE_VIEW AS  
    SELECT      Fname, Lname, Ssn, Dno  
    FROM        EMPLOYEE;
```

## Utilizar a View para inserir dados ...

```
INSERT INTO EMPLOYEE_VIEW values('Julia','Amaral','321233765',2);
```

## Utilizar a View para alterar dados ...

```
UPDATE EMPLOYEE_VIEW SET Fname='Joana' WHERE Ssn='321233765';
```

# View - WITH CHECK OPTION

- Utiliza-se no final da definição da View se quisermos garantir que as condições da cláusula WHERE são verificada na atualização.

## Exemplo

```
/* Exemplo: Vista com os funcionários do departamento 5*/  
CREATE VIEW EMPLOYEE_DEP5 AS  
    SELECT      Fname, Lname, Ssn, Dno  
    FROM        EMPLOYEE  
    WHERE       Dno=5  
    WITH CHECK OPTION;  
  
/* Inserir dados utilizando a vista */  
INSERT INTO EMPLOYEE_DEP5 VALUES('Jose', 'Sousa', 312312323, 8)  
  
-- A operação acima só dará erro se utilizarmos WITH CHECK OPTION
```

# A Seguir?

## Data Operations – Relational Algebra

r		s		r ∪ s	
A	B	A	B	A	B
α	1	α	2	α	1
α	2			α	2
β	1	β	3	β	1
				β	3

$\Pi_{\text{nome}}(\sigma_{\text{farmaco}=\text{null}}(\text{Prescreve} \bowtie_{\text{farmaco}=\text{codigo}} \text{Farmaco}))$

Functional  
Dependencies  
Normalization

## SQL – Data Manipulation

### SQL query:

```
SELECT Pnumber, Pname, COUNT(*)
FROM PROJECT, WORKS_ON
WHERE Pnumber=Pno
GROUP BY Pnumber, Pname;
```

### SQL query:

```
INSERT INTO EMPLOYEE (Fname,
Lname, Ssn, Dno) VALUES ('Robert',
'Hatcher', '980760540', 2);
```

### SQL View:

```
CREATE VIEW EMPLOYEE_DEP5 AS
SELECT Fname, Lname, Ssn, Dno
FROM EMPLOYEE
WHERE Dno=5
WITH CHECK OPTION;
```

# Resumo

- Conceito de View
- Criação e Eliminação
- Actualização de dados via View